USAISEC AD-A268 234

*US Army Information Systems Engineering Command*
*Fort Huachuca, AZ 85613-5300*

U.S. ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION,
COMMUNICATIONS, AND COMPUTER SCIENCES

# AIRMICS

# Visual Knowledge in Tactical Planning:

## Functional Design and Knowledge Representation

## Phase 2 Technical Report



ASQB-GM-91-026
April 1990





AIRMICS
115 O'Keefe Building
Georgia Institute of Technology
Atlanta, GA 30332-0800

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188
Exp. Date: Jun 30, 1986

| | |
|---|---|
| 1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED | 1b. RESTRICTIVE MARKINGS: NONE |
| 2a. SECURITY CLASSIFICATION AUTHORITY: N/A | 3. DISTRIBUTION/AVAILABILITY OF REPORT: N/A |
| 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A | |
| 4. PERFORMING ORGANIZATION REPORT NUMBER(S): ASQB-GM-90-026 | 5. MONITORING ORGANIZATION REPORT NUMBER(S): N/A |
| 6a. NAME OF PERFORMING ORGANIZATION: AIRMICS; 6b. OFFICE SYMBOL (If applicable): ASQB-GM | 7a. NAME OF MONITORING ORGANIZATION: N/A |
| 6c. ADDRESS (City, State, and Zip Code): 115 O'Keefe Bldg. Georgia Institute of Technology Atlanta, Ga 30332-0800 | 7b. ADDRESS (City, State, and ZIP Code): N/A |
| 8b. NAME OF FUNDING/SPONSORING ORGANIZATION: AIRMICS; 8b. OFFICE SYMBOL (If applicable): ASQB-GM | 9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER |
| 8c. ADDRESS (City, State, and ZIP Code): 115 O'Keefe Bldg. Georgia Institute of Technology Atlanta, GA 30332-0800 | 10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. 62783A; PROJECT NO. DY10; TASK NO. 05; WORK UNIT ACCESSION NO. |

11. TITLE (Include Security Classification)

Visual Knowledge in Tactical Planning: Functional Design and Knowledge Representation
Phase I Technical Report

12. PERSONAL AUTHOR(S)

T. J. Hall, J. B. Gilmer, and P. A. Merkel

| 13a. TYPE OF REPORT | 13b. TIME COVERED | 14. DATE OF REPORT (Year, Month, Day) | 15. PAGE COUNT |
|---|---|---|---|
| | FROM 4/1/90 TO 3/30/91 | 1991, April 2 | 30 |

16. SUPPLEMENTARY NOTATION

| 17. COSATI CODES | | | 18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUBGROUP | |
| | | | |
| | | | |

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

A theory of human cognition developed by Professor Lev Vekker holds that human thought involves three distinct modes of processing: symbolic, visual, and kinesthetic, and transformations between these modes. A computer reasoning approach is described that combines the symbolic and visual modes for associate systems. The goal is not only to take better advantage of the human visual mode capabilities for the man-machine system through the improved visualization techniques. The computer will use the visual mode processing as well, using graphic structures and processing normally associated with rendering and image processing as part of the inference process. The technique is initially being applied to military tactical planning, with the selection of an avenue of approach being the initial scenario for prototype development. We envision application of planning intensive activities such as simulation.

| | |
|---|---|
| 20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: [x] UNCLASSIFIED/UNLIMITED [ ] SAME AS RPT. [ ] DTIC USERS | 21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED |
| 22a. NAME OF RESPONSIBLE INDIVIDUAL: MICHAEL EVANS | 22b. TELEPHONE (Include Area Code): 404/894-3107; 22c. OFFICE SYMBOL: ASQB-GM |

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

This research was performed for the Army Institute for Research in Management Information, Communications and Computer Science (AIRMICS), the RDTE organization of the U.S. Army Information Systems Engineering Command (USAISEC). This research is not to be construed as an official Army position, unless so designated by other authorized documents. Material included herein is approved for public release, distribution unlimited. Not protected by copyright laws.



**THIS REPORT HAS BEEN REVIEWED AND IS APPROVED**

s/ James Gantt
James Gantt
Chief, MISD

s/ John R. Mitchell
John R. Mitchell
Director
AIRMICS

# TABLE OF CONTENTS

TABLE OF CONTENTS

## LIST OF FIGURES

# FOREWORD

This report documents the results of Phase II of the Visual Knowledge in Tactical Planning project performed by BDM International under contract number DAKF11-89-C-0069. This work was sponsored by the Army Institute for Research in Management, Information, Communication, and Computer Science (AIRMICS). The report is intended to satisfy the requirements of CDRL sequence numbers A003, Software Requirements Specification, and A004, Software Detailed Design Document. The documentation provided supports the software requirements and design appropriate to a laboratory prototype demonstration planned for Phase III with potential growth to experimental systems in the future.

VISUAL KNOWLEDGE AND TACTICAL PLANNING PHASE II REPORT

## A. INTRODUCTION AND BACKGROUND

This report summarizes the results of Phase II of a three phase program to design and construct a software prototype using a method for computer manipulation and presentation of visual knowledge based on cognitive theory. The specific goal of the program is the design of a software prototype whose displays and behavior are compatible in form and content with the cognitive representations of its users and whose internal representations support direct manipulation of imagistic knowledge.

The planning problem selected for the prototype is the definition of an avenue of approach. An avenue of approach is a route by which a military force may reach an objective. Definition of an avenue of approach is conducted through an analysis based on military aspects of terrain to determine general courses of action available to both friendly and enemy forces.

Phase I of the program was devoted to the conduct of Knowledge Acquisition (KA) designed to elicit from tactical planning experts the knowledge and reasoning processes used to support tactical military planning. Results of Phase I KA sessions, which were presented in an earlier report, included the following:

(1) established the immediacy and importance of visual information in tactical planning;

(2) identified a set of terrain objects and their features and roles that are critical in tactical considerations;

(3) identified a set of map characteristics that inform the perceiver about the terrain objects and their features;

(4) identified a set of composite terrain features that are important in tactical planning; and

(5) captured substantial evidence of the types of reasoning applied to visual information during tactical planning.

Phase III of the program was focused on the specification of the appearance and behavior of the prototype. In addition, Phase II was devoted to the design of knowledge representation that blends symbolic and visual knowledge that will allow the prototype to reason with both modes of information. Phase II of the program will encompass the construct of the

visual knowledge prototype implementing the functional design outlined in Phase II.

## B. TASK 2 - FUNCTIONAL DESIGN OF USER INTERFACE

### 1. Tactical Planning Problem

Users will be presented a military tactical planning map that indicates the terrain features (both natural and man-made) within the area of operations. The map will indicate the current position of a U.S. Army battalion ground unit and the final objective for this unit to be positioned. The tactical planning task is for the user to define the best avenue of approach for the unit to reach this objective by a specified time.

Army training and doctrine manuals were reviewed to identify general rules defining the impact of terrain characteristics on the selection of an appropriate avenue of approach. The criteria include: (1) provision of observation and fire, (2) provision of concealment and cover, (3) avoidance and use of obstacles, (4) use of key terrain, (5) adequacy of maneuver space, and (6) ease of movement. Appendix A lists a set of rules and algorithms obtained from Phase I KA sessions and supplemented by date from military planning documents. These rules and algorithms support an analysis of natural and man-made terrain features for defining an avenue of approach.

The observation and fire criteria are concerned with the effects of terrain on the ability of the force to conduct effective surveillance of an area and the influence of terrain on the ability to engage the enemy with direct and indirect fire weapons. A military planner must consider terrain features that offer their own forces favorable observation and fire. In addition, the planner must consider the effects of terrain on the enemy's observation and fire. Cover and concealment is concerned with protecting units from the effects of enemy fire and shielding of the unit from observation by enemy forces, respectively. Identifying terrain that provides effective cover and concealment often results in an area that has limited observation and fire. Therefore, the criteria of observation/fire and cover/concealment are often in conflict with each other.

The avoidance and effective use of terrain obstacles is another point to be considered in the definition of an avenue of approach. Obstacles are natural or artificial terrain features that stop, impede, divert, or channelize military movement. These include rivers, mountainous areas or ridges, marshy areas, minefields, and anti-tank ditches, to name only a few. An effective avenue of approach should avoid obstacles that are perpendicluar to the direction of the advance and when possible, and position obstacles parallel

to the selected avenue of approach to protect friendly forces against an advancing enemy. The use of key terrain is a criteria that is highly related to other avenue of approach principles and simply states that key terrain should be identified and used to the advantage of friendly forces while advancing through an avenue of approach. Types of terrain features frequently selected as key terrain to support an avenue of approach include a bridge over an unfordable river and high ground to support favorable observation and fire. Adequate maneuver space is a determination of the required maneuver area to support the movement of a given size unit.

The final criteria for the selection of avenues of approach is ease of movement. This criteria includes such factors as relative length of route, direction of route to objective, soil trafficability, and steepness of slopes. This criterian is perhaps the most explicity recognized in the aids and doctrinal procedures of military planning, and may well be the first to be evaluated in practice.

Part of the purpose of planning an avenue of approach is not merely to identify the terrain chosen, but to identify specific problems that the candidate avenue presents. These problems may require special resources such as smoke, fire support, bridging equipment, or other engineering support. These needs resources can be compared with resources available, or additional resources can be sought. No avenue of approach is perfect. For example, the "best" avenue will usually be the most heavily defended. Selection of an avenue of approach is a matter of making tradeoffs in desirable attributes and resources needed. Recognition of a seemingly unsuitable avenue as practicable is often the mark of a good plan. As an example, the German use of the difficult but poorly defended Ardennes Forest in 1940 as an avenue of approach for their main attack was decisive.

Weather conditions interact with terrain to influence the selection of appropriate avenues of approach. Different weather conditions affect surface condition of terrain, thus, substantially influencing the trafficability of various areas. For example, snow and ice may make steep grade areas impassible that would otherwise provide an adequate route. Rain may cause a clay road to become impassible. On the other hand, rain may make a sandy road more firm and, thus, more easily crossed. Fog may provide cover and concealment along a route that otherwise would be vulnerable to enemy fire.

At a more general level, time of year, or season may influence the selection of an avenue of approach. While deciduous trees provide excellent cover and concealment during the spring and summer season, the

loss of their leaves in late fall and winter no longer provides cover from enemy forces.

## 2. User Displays

### a. Basic Terrain Map

A basic military terrain map will serve as the baseline display for the prototype. This map will contain color and standard map symbols to depict significant man-made and natural topographical characteristics of the area, such as transportation networks (e.g., railroads, autobahns, roads), vegetation (e.g., deciduous forest, coniferous forest, swamp), bridges and hydographic features (e.g., iron bridge, wooden bridge, canal), relief feature (e.g., contours, steep slope, rock), and other miscellaneous topographical details (e.g., embankment, tree lined roads, spot elevation).

A standard military terrain map will be input to the computer using a color scanner. The geographical area selected for the problem is Germany, specifically the area surrounding the Fulda Gap. This area was selected because it is an area rich in terrain features. It also is an area that is frequently used in military training and studies. Thus, a substantial amount of information is available related to the characteristics of the land and the military significance of these features. The location of significant map features will be defined as objects using a tool such as Adobe Illustrator. The map will indicate the current position of a U.S. Army battalion ground unit and the final objective for this unit to be positioned.

### b. Overlay of Specific Go/Slow Go/No Go Terrain Areas

Since the purpose of the avenue of approach is to provide a corridor for movement, trafficability is an important consideration. An avenue of approach ideally implies enough flexibility and redundancy of potential routes within the approach that any single event will not have a significant effect on movement of the unit as a whole. Thus, trafficability is evaluated on a regional rather than point or route basis. The formal step in initially evaluating an area of terrain for an avenue of approach is to designate on a map various regions as go, slow-go, or no-go terrain. Regions of no-go terrain can be tolerated within the avenue of approach on a trafficability basis as long as they can be bypassed, and do not cause a bottleneck. Where bottlenecks do occur, such as single bridges or mountain passes, these would be listed as potential problems with the candidate avenue of approach.

### c. Overlay that Smooths the Edges of Go/Slow Go/No Go Terrain Areas Displaying General Trafficability Conditions

Classification of individual array elements as go, slow-go or no-go terrain typically, provides more detail than necessary when considered against the rather broad area needed for an avenue of approach. A logical connecting step between classification of Go terrain and the identification of potential avenues is to "blur" the array of trafficability classifications. This will be a process very similar to that used in image processing, and can be considered a "low pass filter" on the data. Typically, a calculation is made for every element in the array of trafficability values for the region. The value for the "blurred" image is a weighted sum of all of the array elements within some number of points. The resulting array has just as many points as before, but the edges are smoother. Single points of good trafficability in regions of mostly poor trafficability will tend to disappear, as will points of poor trafficability in regions of good trafficability. (An important consideration which may require special treatment is that linear features such as rivers and escarpments that are important but very narrow are not blurred out). The blurred image can then be thought of as having a much lower complexity. The features of the blurred array can be extracted, including regions of good trafficability that are candidates for avenues of approach and region of no-go that act as important barriers to any corridor of advance for threat forces.

### d. Areas Vulnerable to Enemy Fire - Line of Sight Calculations

An important consideration in defining an avenue of approach, is the degree to which a force moving along the route will be exposed to enemy fire. This is a much more complicated consideration than trafficability. It requires reasoning of the following sort: "If I move down this potential avenue of approach, from what potential enemy positions can I be seen and engaged?" This display can be constructed by identifying all high elevations which appear on the map, then exploring the lines of sight available from each to see what the intersection is with a candidate approach.

This display also can provide information inverse to areas of vulnerability to enemy fire. Line of sight of the unit marching through the avenue of approach can be displayed for selected points along the route.

#### e. User Selection of Avenue of Approach

Users will be able to draw one or several candidate avenues of approach onto the basic terrain map. These candidates routes can be drawn on either the basic terrain map alone or in conjunction with either of the go/slow-go/no-go terrain overlays. The user will input the selected avenue of approach with a "point, click, and drag" technique with a mouse.

#### f. Alternative Avenues of Approach

The prototype also will have the capability to generate possible alternative avenues of approach other than the route(s) selected by the user. Given the line of departure, the final objective, size of the force, weather and season effects on terrain, and the timeframe to reach the objective, the computer will generate and display alternate avenues of approach to be considered by the user.

#### g. Time/Distance Analysis of Alternate Routes

Time to reach the objective is a critical consideration in the selection of an avenue of approach. Not only is the total travel time of importance, but the time if takes to cross specific sections of the approach, especially high vulnerability areas, should be considered. A time/distance display will assist the user in the evaluation of a single or alternative avenues of approach by simulating the progress of the advancing forces through selected routes. The display will show the starting point of the unit and then will track through a single route or simultaneously through multiple routes with relative rates of advancement calculated by the computer. Advancement calculation will be based on algorithms related to the size of the unit and characteristics of the terrain along the route (e.g., type of road surface, width of road, slope of terrain). This display will demonstrate the effects of Slow-Go terrain and bottlenecks on the trafficability of the avenue of approach. Vulnerability to enemy fire also can be presented in the display to highlight areas along slow go routes or bottlenecks that would extend the period of time that the unit is vulnerable. This display can be generated for a single selected avenue of approach or used as a means of comparing alternatives through simultaneous presentation of routes.

Key points along an avenue of approach such as bottlenecks and points of high vulnerability to enemy fire will be tagged during this analysis. These key points or events will be marked on the display using arrows. In addition to marking key events (e.g., bottlenecks, high vulnerability areas) along a candidate avenues of approach, a timeline will be constructed

for each avenue of approach with an indication of the point along the timeline that a key event occurred.

#### h. Cross-section View of Key Areas Along Route

Key events highlighted along an avenue of approach can be selected from either the time/distance display or the timeline display. When a key event is selected the user is presented a horizontal cross-section view of the terrain associated with the event. This provides a different visual perspective of the problem to the user. This analysis supports the user in deciding whether to selected an avenue of approach that crosses this terrain areas or to define tactical measures to deal with the hazards associated with the event.

## C. TASK 3 - KNOWLEDGE REPRESENTATION FOR THE INTERFACE

### 1. Computer Representation and Manipulation of Visual Knowledge or Conceptual Graphs and Visual Knowledge

One of driving principles of Dr. Vekker's theory is that in human cognition, symbolic reasoning rests on a foundation of visual reasoning. This is comparable to the fact that in computers, a higher level language may be used, but it actually is expressed in the machine in the more primitive and less expressive machine language. This implies that symbolic knowledge in our representation should rest on, or at least be linked to, visual knowledge. When symbolic knowledge is being manipulated, images should be associated and driven in a manner to follow the thread of symbolic reasoning, and where appropriate, contribute new meaning that would not arise from purely symbolic reasoning. Figure 1 illustrates the approach for representing the symbolic and visual modes, and the links between them in the development of the prototype.

At the top of Figure 1, is a representative concept instance and its type, as represented with Conceptual Graphs. The "concept" might be an object such as a military unit or a hill, a property such as speed, or a more abstract concept such as a network of roads or organization of units. Associated with the concept and its type is an icon, a graphic representation of the object. The icon may be an attribute of either the concept type or the concept instance, inheritance being used in the latter case. The icon actually may be determined based on a number of attributes of the instance and may be generated when needed. This would be true, for example, for a visual image generated to represent a person.

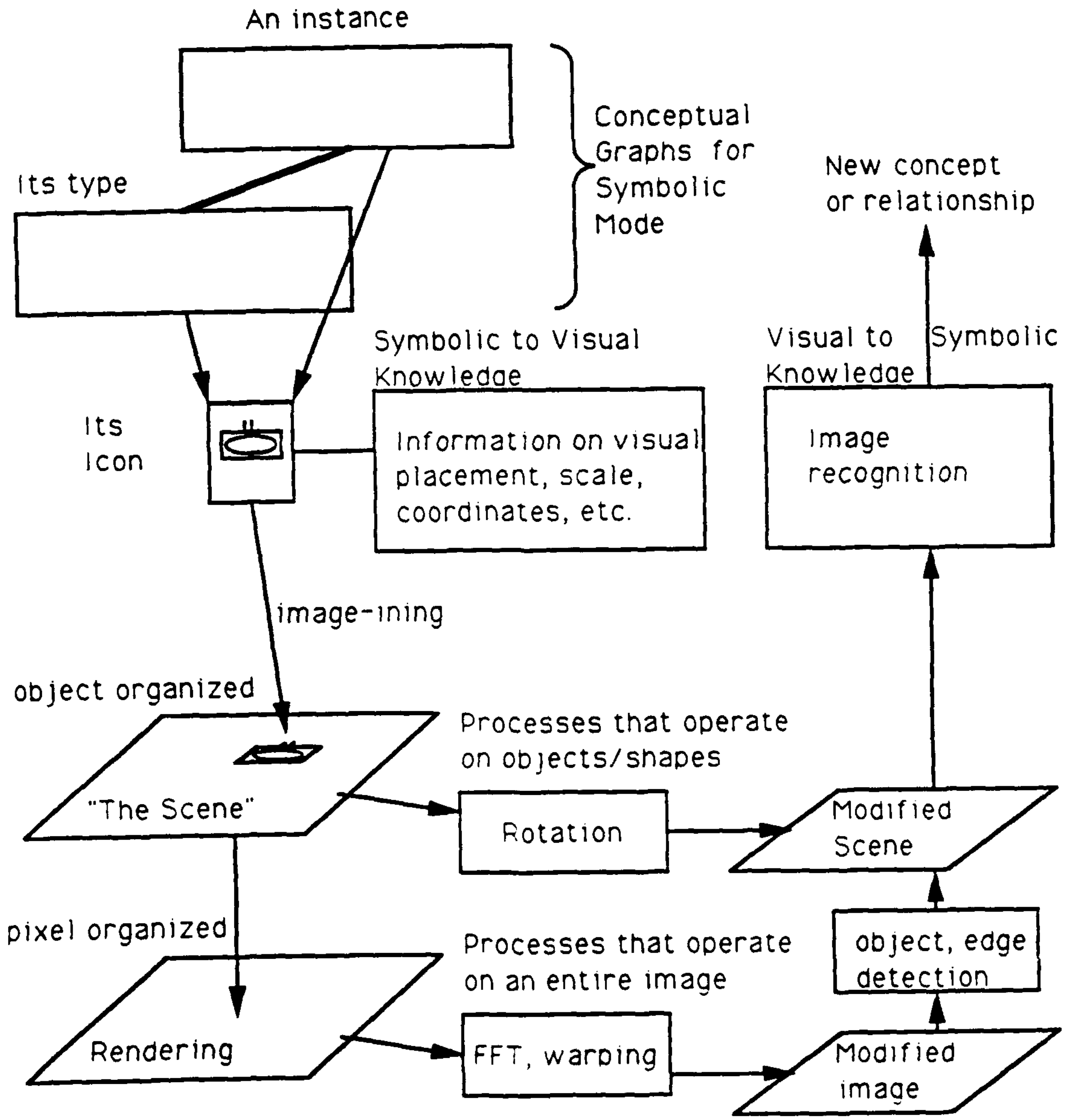


Figure 1: Representation of Visual and Symbolic Modes

The icon and its generation is part of the process of generating a visual mode representation of the symbolic mode context. The icon is mapped into "the Scene," which may be thought of as the visual counterpart to the context. Here the term "the Scene" is meant more in the sense of the colloquial term, or drama, implying a sense of the entire context, not merely a view of it. In this sense it is object oriented, with structures representing various entities. However, unlike the symbolic domain, relationships are represented in spatial placement rather than in connections.

In order to map the icons to the scene, knowledge about making transformations from the symbolic to the visual is needed. A very simple case occurs when the symbolic objects have location attributes in the same dimensional space as the scene. For example, military units having UTM coordinates can be readily and straightforwardly mapped to a "scene" organized as a map. Only the scaling of the icon, choice of how to represent auxiliary information such as the unit's identity, and transformation from the UTM coordinate system to the map's x, y coordinate system is necessary.

Things become more involved if the attributes do not map so obviously. For example, if the instances have attributes in three dimensions and "the scene" is two dimensional, some projection must be chosen. If the attributes are more abstract, such as some purely in terms of relationship, a more complex mapping is necessary.

Consider, for example, an electronic circuit. Components have relationships that are graphlike, and represented in such terms in the Conceptual Graph. However, a mapping to "the Scene," which in this case would be schematic, has too few obvious mapping principles or reference points in the concept graph proper. Instead, auxiliary knowledge must be invoked. For schematics, this would include such things as a preference to have higher potentials at the top of the page, a preference for signal flow (which must be reasoned about itself) go from left to right, and for the icons of certain components such as transistors to have a specific orientation. PERT diagrams, organization charts, program flowcharts, and other visual representations of more abstract data also would have their own sets of rules. Knowledge for which no preconceived notion exists would map presumably to a more generic form, perhaps a link-node type of graph.

Operations taking place in "the Scene," could be characterized as being "visual," although the manipulations are oriented around objects. Such operations could operate on the scene as a whole, such as a rotation in perspective, or on particular objects, such as coincidence (or overlap) detection. The manipulations in which time is advanced could be considered a

form of model-based forward reasoning or simulation, since the characteristics of particular objects plays an important role.

It could be argued that all of this is merely another form of symbolic reasoning, and that all of these manipulations could be represented in a Concept Graphs framework. Perhaps the essence of the difference is that in the visual mode relationships are portrayed spatially, and that the human apparatus includes special hardware to support mental spatial manipulations. This would include model-based reasoning for physical objects that are moving and position comparisons without having to resort to explicit reference to a symbolic form of object properties. This distinction is a bit artificial in the computer, which at its foundation is a symbol manipulator. On the computer, these "visual" operations must be implemented with code that looks very much like symbolic reasoning, even if the motivation and high level design is different.

It is likely that these visual manipulations will map somewhat better onto highly parallel machines, specifically SIMD types, than most general forms of symbolic reasoning. Subjectively, human symbolic reasoning seems a serial or near serial process, while human visual processing is clearly parallel. It may be that human leaps of insight which seem to involve parallelism actually use the visual machinery, but on abstract representations of the symbolic problem. This would seem consistent with Dr. Vekker's theory on visual knowledge. Similarly, in a computer environment the more organized nature of operations in the visual domain may allow mapping to less general but more powerful machinery. Such a mapping to specialized machinery would seem to satisfy the need to show a distinction, even though the same basic computational building blocks are used.

Underneath "the Scene," the figure shows a rendering that can be thought of in computer terms as a raster array of the scene as seen from a particular perspective. In general, this area may not be limited to two dimensional. In transforming the scene information to such a pixel array form, much information about specific objects may be lost. What is gained is an ability to "see" the scene in a holistic manner and organize its contexts differently. Thus, an assemblage of individual objects may be recognized as constituting a whole that could not be easily inferred from the pieces. For example, trees become a forest. The recognition of such aggregate objects would depend on the traditional tools of image processing: smoothing, thresholding, edge detection, transforms, and finally, object recognition. The newly recognized object becomes a coherent whole in "the Scene," absorbing or connecting the separate parts that were present but previously not

associated. The object also maps into the symbolic mode as a new concept, with specific relationships to other concepts.

The process of being able to construct such wholes from parts, depends greatly on conditioning of "the Scene" prior to rendering. For example, by differently coloring shapes that are observed to be in motion in different directions, one may discern more easily which objects may be parts of the same whole. Choice of perspective also is important. An edge-on view of the battlefield may be useful on some occasions, such as where sectors are narrow, attacks are frontal, or it is important to appreciate the role of altitude in the air war. However, such a view does not permit an appreciation of ground maneuver, especially flanking attacks. Seen edge on, such battles make no sense. Thus, visual operations on the scene in the object oriented sense are a vital complement to the rendering, pixel oriented recognition process. What you can infer from the scene depends on how you see it.

It seems reasonable that meta-information may itself be visual or symbolic. Such information would be organized around canonical types of scenes. These would include perspective views, maps, flowcharts, hierarchy charts, and presumably many others. Each would have knowledge about how to map icons to the space for the particular "Scene," and suggestions on different transformations or coloring to consider in order to make the whole meaning of the scene more apparent.

This further development on the representation and processing of visual information would seem even more distinctive from the symbolic mode than those of the object oriented manipulations mentioned earlier. If we can implement a sufficient portion of these in our prototype, we should be able to show the distinctions and advantages of the use of visual knowledge effectively.

### 2. Selection of Software Prototyping Tool

To support selection of the software prototyping tool, candidate implementation tools that would meet cost and effort constraints during prototype development while having the necessary technical capabilities to support the program goals. Based on the objectives of the Visual Knowledge program, the following requirements were formulated for the prototyping tool:

#### a. Support an Object Oriented Representation:

This requirement is desirable for both software engineering and representational reasons. Support for object hierarchies and messages is expected to allow a more powerful and less expensive approach,

with rules, entities, relations, sets, and other items about which the system must reason represented as "objects."

**b. Support the Representation of Images, Icons, and Graphics:**

This requirement is necessary to support the visual elements of the prototype. There are insufficient resources in the contract to support development of these features from "scratch". Features to support creation and manipulation of graphic entities, therefore, must be provided by the selected programming tool.

**c. Support Visual Interaction with the User:**

The graphics features referenced above must be part of the user interface. This will allow the user to select points and items (e.g., with a mouse) that the system will relate to various graphic and iconic information. This task will require a minimum of programming effort.

**d. Minimal Programming Effort:**

Contract resources do not allow for a significant level of effort to support extensive programming. A tool with a clear and intuitive programming metaphor is needed. General purpose programming, using C, C++, or PASCAL for example, may be useful for selected special instances. However, such programming is too labor extensive to be used for the entire programming effort.

**e. Support Needed Computational Operations:**

This criterion was viewed as the degree of flexibility and power provided by the tool.

**f. Inexpensive to Acquire and Operate:**

The acquisition and computer operations ceiling costs were set for a few thousand dollars, given the budget for the prototyping effort as a whole. This implies the use of microcomputer-based software rather than workstation-based software, since acquisition costs for the latter are usually significantly higher and the time on workstation is billable and expensive relative to contract resources.

Given the availability of the Macintosh IIfx at BDM and the relatively advanced state of Macintosh software for user interface and image manipulation, a Macintosh approach was selected for the prototype. Candidate software tools considered for the project included the following:

(1) Hypercard Claris (Apple) (Free) - Hypercard comes with every Macintosh computer. The major disadvantage of this program is that it does not support color or graphics objects in a programming sense. BDM has experience programming with this tool.

(2) Supercard Silicon Beach Software ($200) - Similar in many respects to HYPERCARD, this environment is oriented more toward "engineered" products than the relatively unstructured Hypercard. It supports multiple, variable sized windows, color, and animation. It can use the same interfaces to external programming languages, XCMD's, as Hypercard. BDM has experience programming with this tool.

(3) Prograph TGS Systems ($195) - The "guts" of an application using Prograph still is programmed in C, but the interactive aspects are "visually" programmed. The program appears to support all of the features required, but would likely involve more programming effort than either Hypercard or Supercard.

(4) Serius Developer 2.1 Serius Corp. ($495) - This is a programming tool seemingly more comprehensive than Supercard or Hypercard. The tool is built around object oriented and visual programming principles. A review of the program published in MacUser March 1991 indicated the possibility of significant effort needed to reorient programming habits, though this is true likely for any of the choices. However, BDM does not possess experience specifically with this tool.

(5) Iconix Power Tools Iconix ($thousands) - This tool set is intended to bring object oriented and iconic design principles to software development, and is organized around industrial strength programming support rather than prototyping. Iconix also has an Ada product. The price and orientation seem to put this product beyond the bounds of the requirements. A demonstration disk of this tool was reviewed.

(6) MacObject LPA ($500) (requires use of MacProlog, also $500) - This tool is more of a general AI type

environment. Numerous Lisp, Prolog, and expert system tools also are available, but are of unknown utility. These choices were not investigated in any depth.

(7) MacApp Apple ($100+) - This is a programming system for developing applications based on the use of PASCAL. or C++. It requires a good bit of programmer reorientation and investment, and is not really visually oriented, though it provides tools for developing Mac standard types of dialog boxes, displays, etc.

In considering the programming tool choices, it was not possible to obtain a comprehensive look at some products, such as Serius Developer 2.1, for which no software was owned or demo available. Examination of the choices proceeded from those perceived as easiest to use toward those more difficult or less accessible. Hypercard was considered unsuitable due to its inability to support color displays and graphics objects as program elements. Supercard was examined next and was determined to be suitable as a software tool. The only reservation about Supercard is that the computation speed for executing the interpreted "Supertalk" language is slow. For a prototype, this should not be a primary consideration. Supercard does provide for the use of external compiled code (XCMDs and XFTNs) which allow selected functions to be developed in compiled C or PASCAL.

**g. Supercard Description:**

Supercard defines a number of objects: graphics, buttons, or files are the lowest, most detailed objects. These are on cards that are the basic elements of display. Multiple cards can share a background. An assemblage or stack (to use Hypercard terminology) of cards are associated with a given window. A project contains perhaps several windows. Above that, a shared resource pool may be used, with several "projects" open at a time. Messages are passed up this hierarchy, so that a behavior to respond to a message or mouse click is passed up until a handler is found. This is the manner in which inheritance is implemented in Supercard, as shown in Figure 2. In this form, Supercard is rather rigid and does not exactly suit the programming needs for the prototype. However, the "pass" and "send" commands of Supercard should provide the needed flexibility. The problem is to determine how best to use these facilities.

The three basic things to represent within the Visual Knowledge prototype are entities, their attributes, and relations. It seems clear that an entity, such as a mountain, road, military unit, or other such tangible object

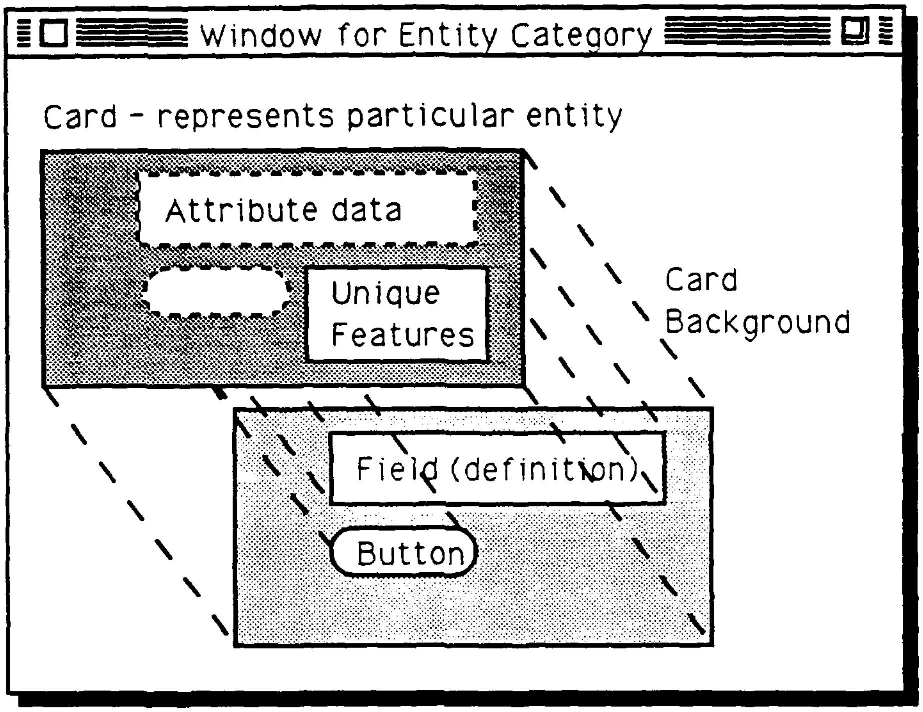


Figure 2: Basic Organization of Objects in Supercard

should map to a Supercard card. Entities of the same type, thus having a common inherited behavior, would logically share the same background. In fact, the background would contain most of the structural information, such as field/attribute definitions, with the cards having only the data that differs from one such entity to another. Figure 3 illustrates this approach.

All entities will be represented on separate cards. The map itself (or the several maps) will be individual cards. Map objects also will be separate cards. The graphics on the map itself would be merely renderings, or displayed manifestations of the the actual entity. Figure 4 shows this arrangement.

Attributes of entities are in their simplest sense quantities or other parameters having a fixed meaning, but being variable with respect to different entities. While the values of such an attribute must be placed at the Supercard card level, since a card is used to represent an individual entity, the

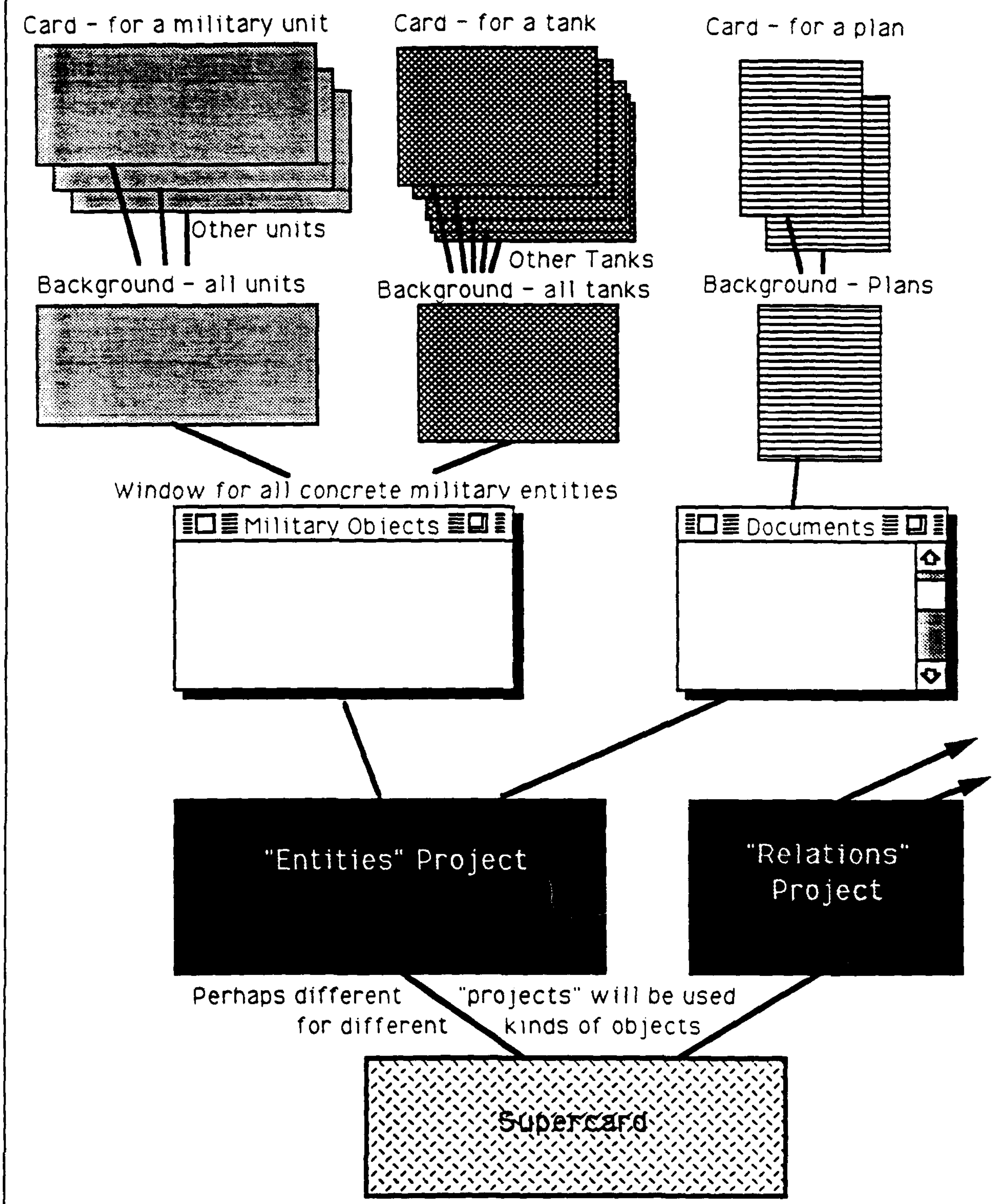


Figure 3: Mapping of Entities to the Supercard Object Heirarchy

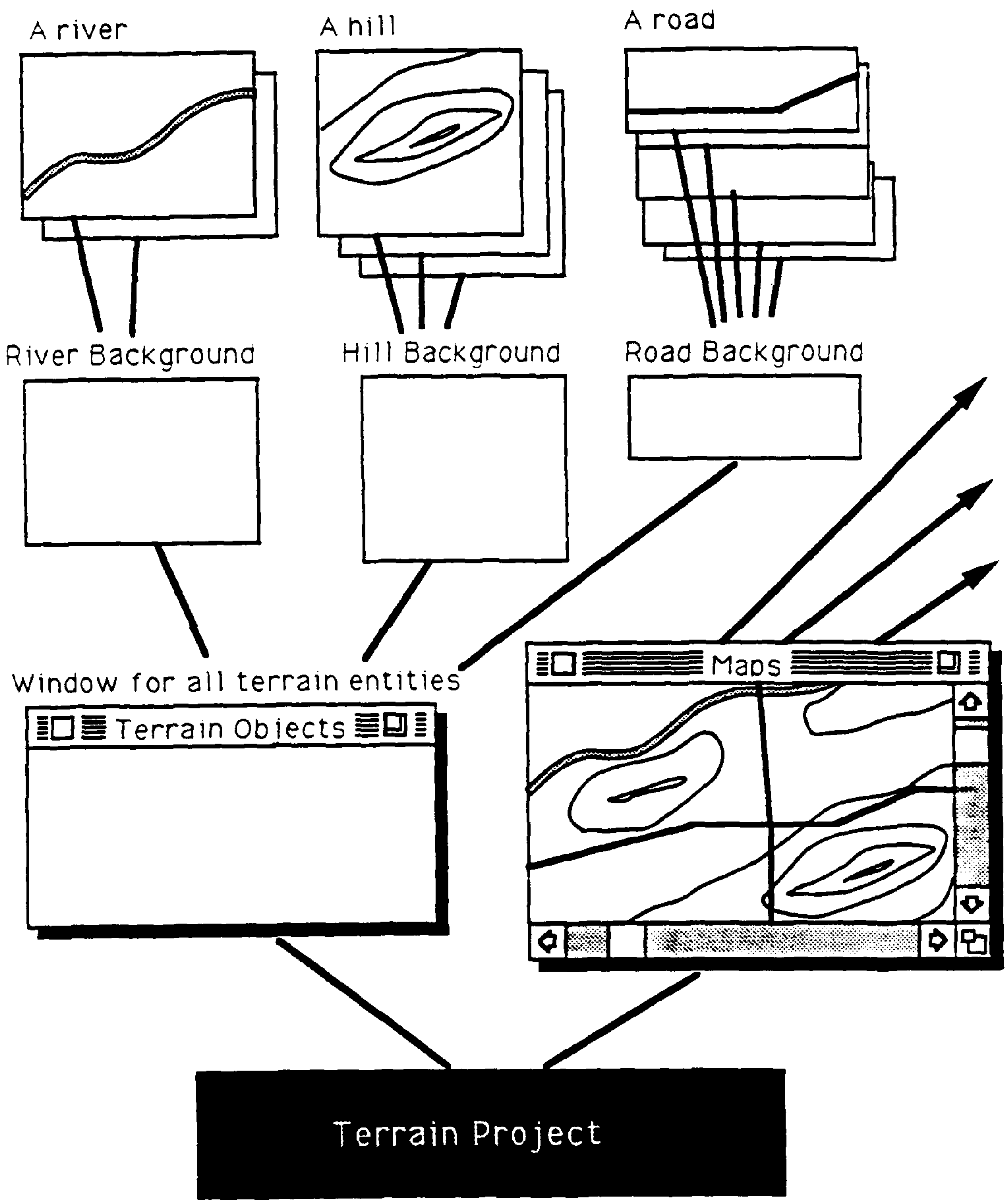


Figure 4: Representation of Terrain Objects and Maps

attribute itself would logically be defined as a field at the background level. To a user, the location of the text datum would be fixed for all entities sharing a background.

For this project, it seems reasonable to use a card for defining characteristics of an attribute. The card would have the same name as the field on the card of an entity having the attribute. This results in a consistent means for accessing attribute data by the reasoning engine. The reasoning engine, or components of it dealing with attributes, would logically be scripts of the attribute backgrounds and window. Figure 5 illustrates the arrangement of Supercard objects for attributes.

The other very important component of knowledge to be represented is the set of relationships. Where an attribute of an object generally will be a single quantity without direct reference to other objects, a relationship will connect two or more objects. A simple means of representing a relationship is exactly as for an attribute, but replacing the quantitative value with the identifying name of the referenced entity. This approach is limited to relationships that have been anticipated for the given entity, while in fact many relationships will very likely be constructed dynamically.

Dynamic relationships may exist during the course of reasoning about the situation. Given that the relationship is a focus of reasoning, it makes sense to treat it fully as an object. All relations of a given type would presumably share the same background, in a window dedicated to relationships. The individual relationships would thus have attributes, such as degree of truth or constraints under which the relationship holds, just as do other objects. Figure 8 presents an illustration of such a dynamic relationship.

Note that some relationships, specifically the type and classification of an object are built into Supercard by having objects of the same type share a common background and window. This is an implicit relationship. Altogether there may be three different ways for representing relationships: implicit (using Supercard's inheritance), relatively static (using fields), and dynamic (using relationship cards).

### h. Viability Test of Supercard

A simple viability test was performed to validate the appropriateness of Supercard as the software environment for the Visual Knowledge prototype. A small feasibility demonstration of prototype capability was used to test the ability and flexibility of Supercard for representing visual knowledge. Examples of rules, graphic objects, and symbolic representations of units were tied together and demonstrated.

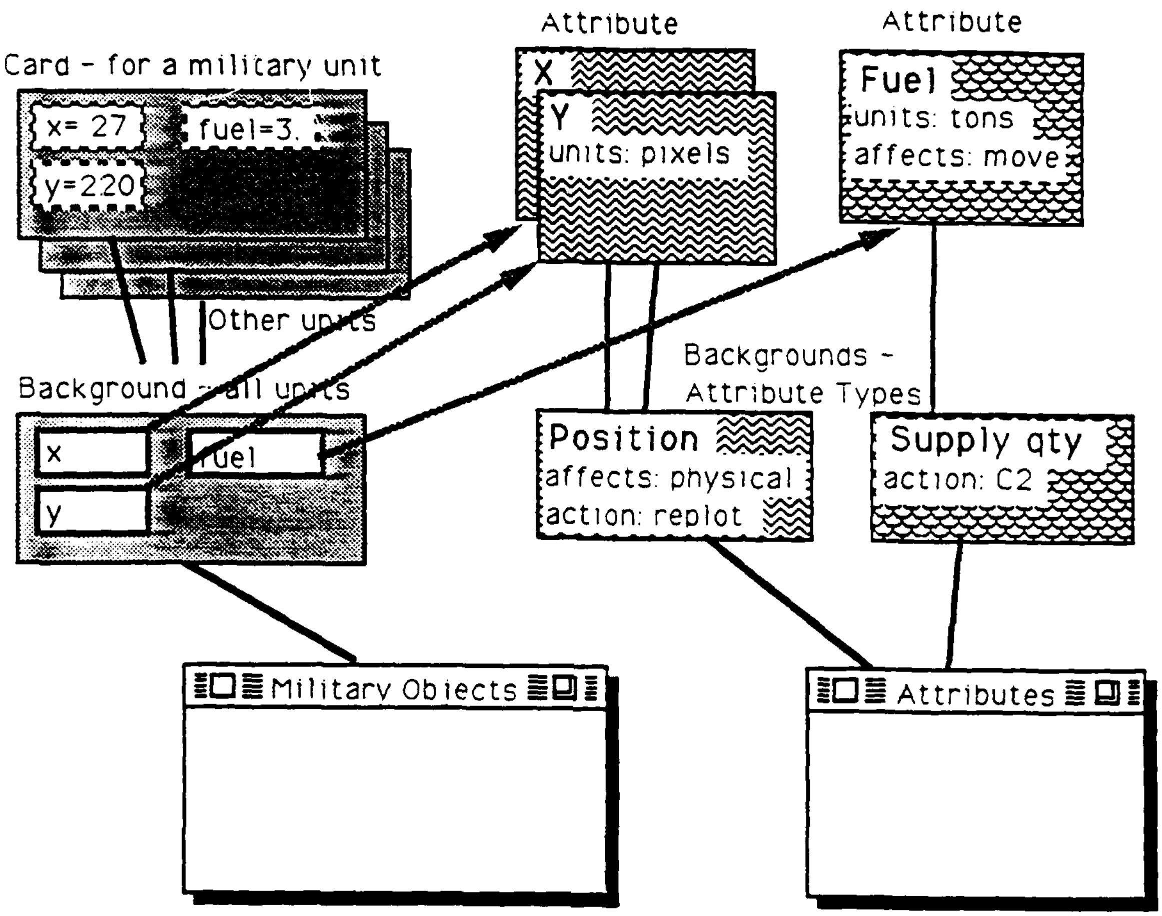


Figure 5: Representation of Attributes as Objects

In the course of examining Supercard, it was necessary to develop preliminary software representation schemes for the symbolic and visual knowledge to be represented. It was found that the graphic item, card, background, window hierarchy of objects in Supercard maps does provide

display and inheritance mechanisms that will be needed to support the prototype. It is anticipated that separate windows will be used to represent military units (symbolic information), maps, rules, attributes, and other kinds of objects. This will allow the user to have simultaneous access to multiple representations and the connections between them.

The functions demonstrated by the prototype include line of sight calculations based on graphics objects, simple calculations based on direct access to the pixel oriented rendering of the scene, and animated movement of objects on the map for depicting trafficability effects. The conclusion reached from the test was that Supercard will meet the prototyping needs as expected.

## D. COMPUTER REPRESENTATION AND MANIPULATION OF SEMANTIC AND VISUAL KNOWLEDGE USING SUPERCARD

The basis for representing symbolic knowledge will be Sowa's Conceptual Structures, described in "Conceptual Structures, Information Processing in Mind and Machine," by J.F. Sowa, Addison-Wesley, 1984 and Berg-Cross and Price in "Acquiring and Managing Knowledge using a Conceptual Graph Approach," IEEE Transactions on Systems, Man and Cybernetics, Vol. 19, No. 3, May/June 1989. Conceptual Structures has three basic types of objects: referents (or instances), types associated with the instances, and relations. The instances and types appear to map well to Supercard cards and backgrounds, with a window being designated for a variety of types of concepts (objects). Relations seem also to map well to cards. Backgrounds for such cards would logically correspond to types of relations, with particular cards being the specific relation associated with a given instance. Figure 6 illustrates how this would map to the Supercard object hierarchy. In Conceptual Structures, "attributes" are just another kind of relationship. In Supercard, attributes would map well as fields. It may be appropriate to diverge from the purely relational approach described earlier for attributes to keep the number of cards down and to allow the attributes of an instance to be more easily viewed. There are some other practical tradeoffs to be examined as well, specifically in how to represent fuzzy, indefinite, or probabilistic attributes and relations. It will not be practical to implement a complete Conceptual Structures system in the context of this project. As the scenario becomes defined better, it will be possible to select those features for implementation in general form, and others to implement only in a form that works for the scenario but without the full generality of Sowa's system. We do not expect to implement a full, general purpose reasoning engine, for example.

The manner of operation envisioned for a system using visual knowledge is illustrated in Figure 7. The front end is shown in this case as a

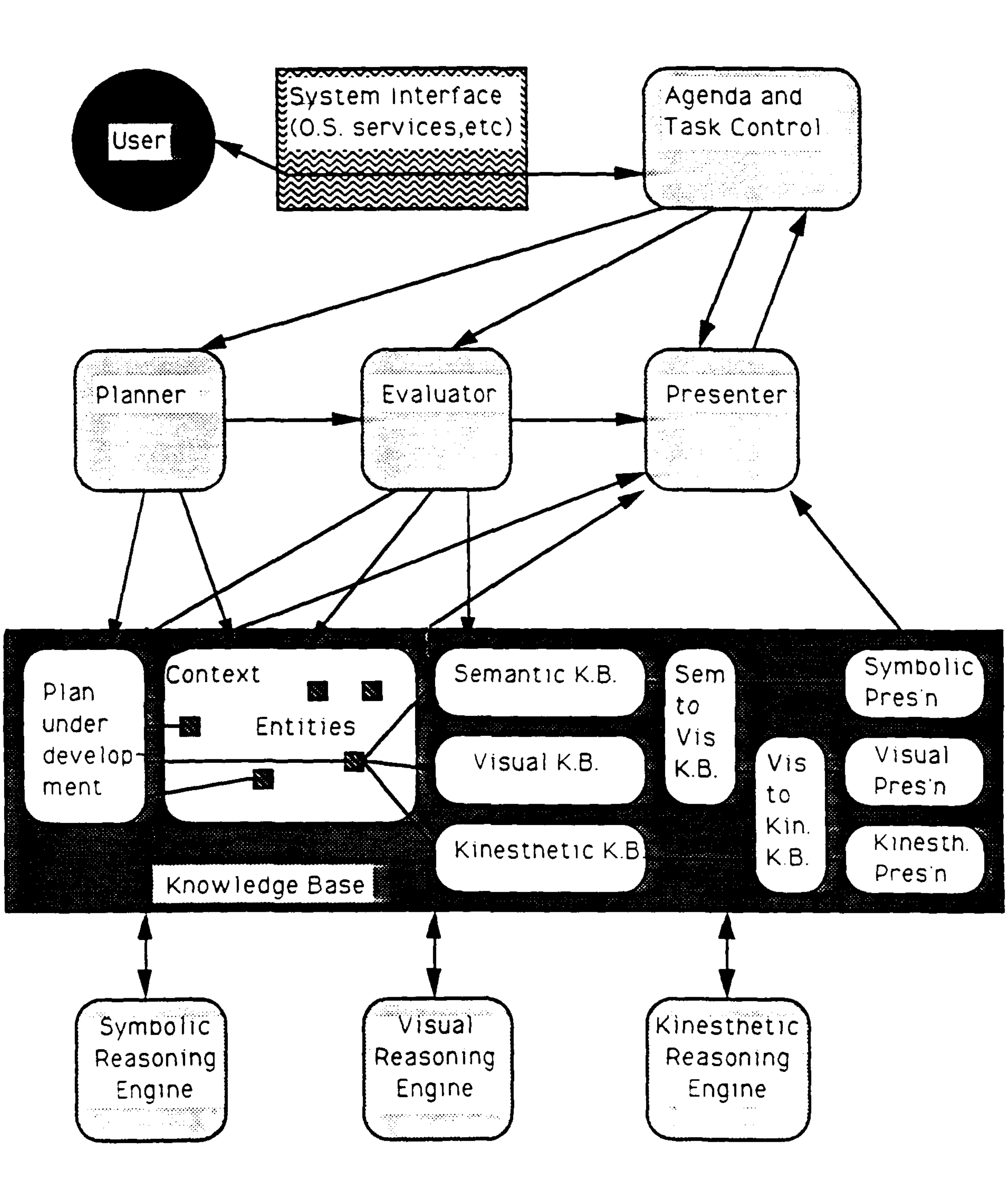


Figure 6: Proposed Support System for Combat Simulation Operation Plan

query in a natural language that is translated into symbolic form. The actual system would incorporate visual and button types of interfaces for most

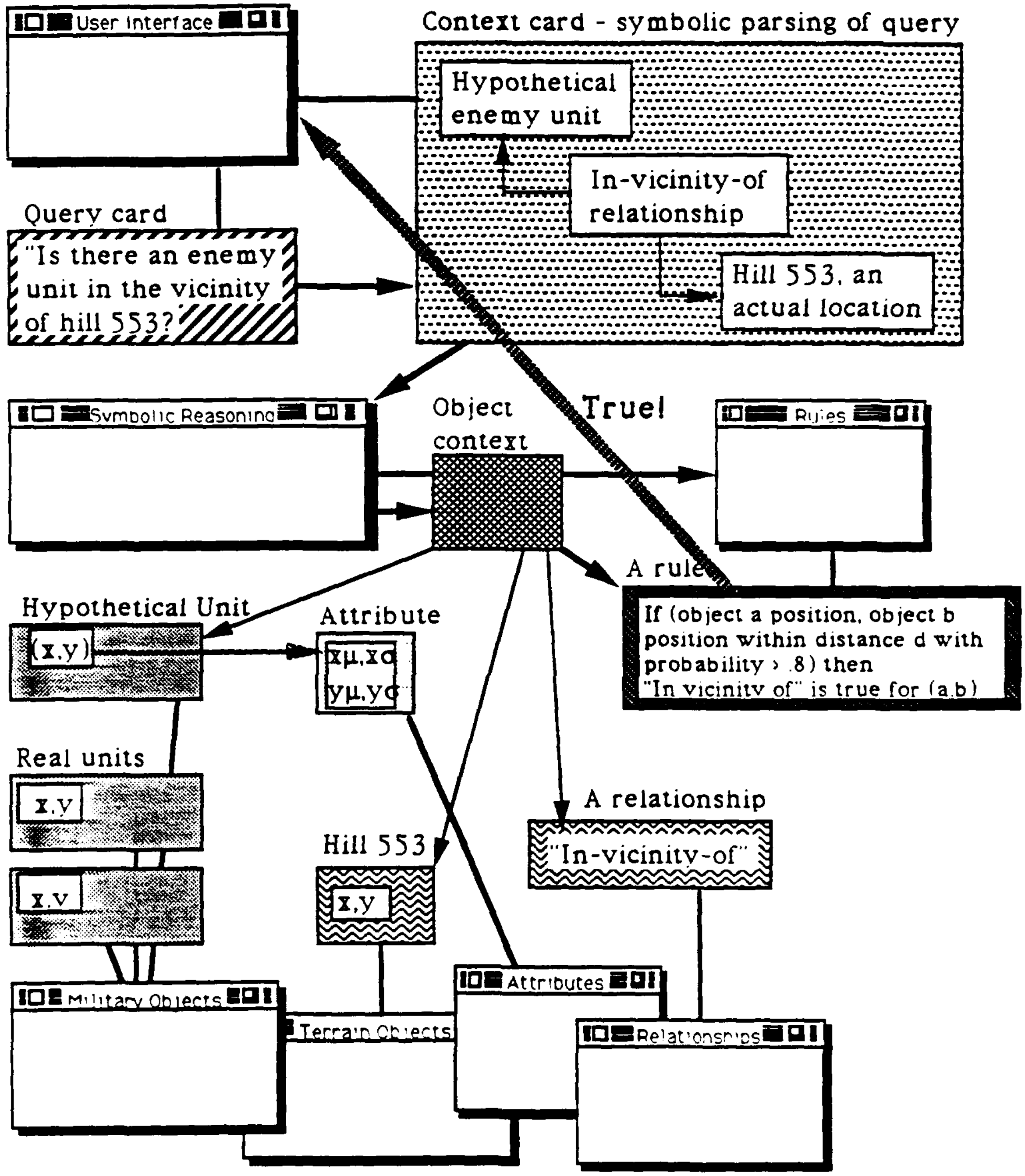


Figure 7: Symbolic Knowledge as Conceptual Structures in Supercard

queries. This figure serves only to illustrate the mapping to Supercard for symbolic knowledge, which is the departure point for the visual knowledge

representation. The boxes represent Supercard entities for cards, their fields, backgrounds, and the windows in which the cards appear.

As described in section C.1, incorporation of Visual Knowledge using the computer seems to require two separate treatments. In one sense, the visual scene is made of objects that are portrayed in space but retain their identities and are separately and individually manipulable. In another sense, objects merge together to form patterns which, in a computer, seem best represented by an array of pixels. These two methods correspond to distinctively different commercial software packages for producing images; "Paint" and "Draw" programs. The former are most appropriate for artistic work where the composition of the whole is most important. The Draw programs are most appropriate where precision and distinct component identities are important, as for engineering and architecture. Supercard incorporates both of these constructs.

It is envisioned that both of these treatments of visual knowledge will be incorporated into the prototype as depicted in Figure 8. Some forms of visual reasoning concerning the relative placement of objects, scaling, and other basic object characteristics and relations might be resolved at the object oriented level for describing a scene. However, it is expected that the most powerful and unexpected (in symbolic terms) results will occur when the scene is rendered as a bit map of raster form, and a new organization (set of objects) is recognized which may lead to conclusions hard to reach without "seeing" the problem differently. The thread of reasoning will proceed using both visual and symbolic modes, as shown.

Within Supercard, the "Scene" will be a card on which various objects, symbolically represented as cards, will be presented as discrete graphics. In Supercard, such graphics all have properties, can be "clicked on" using a mouse, can be manipulated (scaled, etc.), and can send and receive messages like other objects such as cards. Functions will evaluate object relationships such as whether objects overlap, distances between objects, etc. The Macintosh toolbox will handle graphic rendering, producing the pixel patterns that appear on the screen. Access to these pixel images is possible with C through the "External Commands" feature of Supercard. These external commands and functions access but do not change the screen pixels. External commands and functions may copy screen pixels and perform operations such as blurring, convolutions, and other manipulations typical of image processing. When moving back to the object-oriented and symbolic levels, thresholding and identification of edges, graphs, etc. will allow a new set of objects to be created, in a manner developed by the image recognition community. Only a

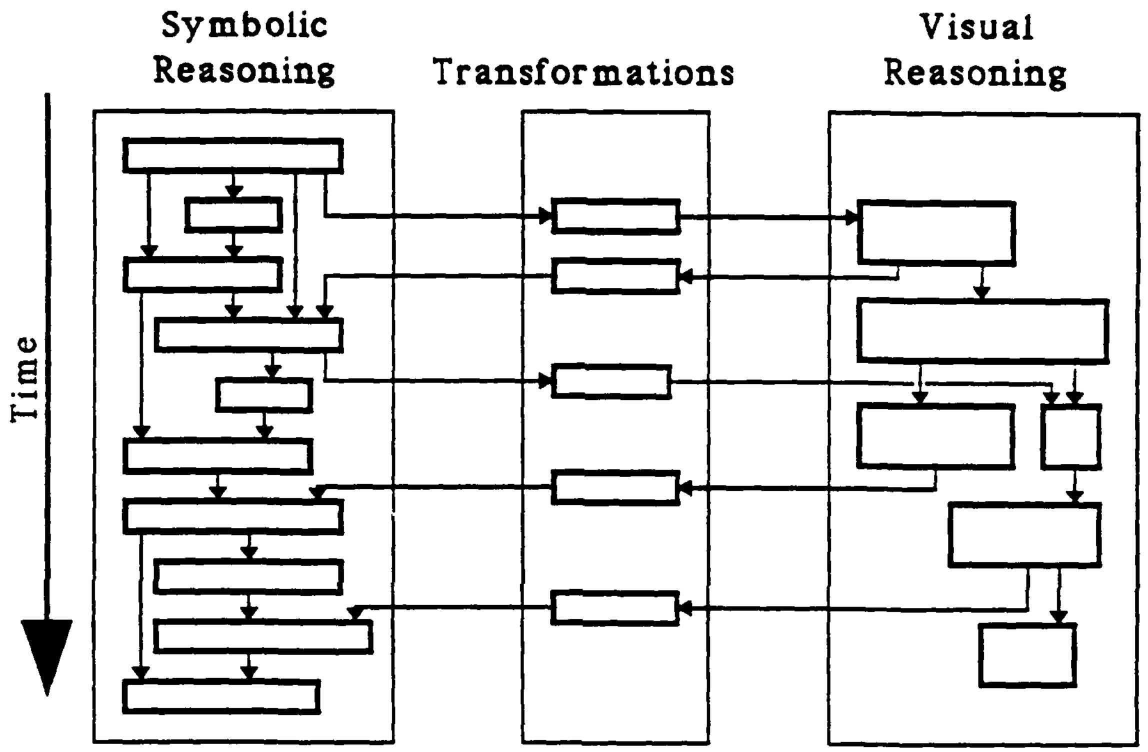


Figure 8: Integration of Symbolic and Visual Knowledge

few representative and illustrative examples of such processes will be implemented in the prototype to support the scenario problem.

## E. PHASE III GOALS AND OBJECTIVES

During Phase III, BDM will implement the designs of Phase II outlined in this report in a proof-of-concept software prototype of an interface for defining avenues of approach. The prototype will embody sufficient functionality and intelligence to demonstrate the possibility and value of building cognitively compatible user interface for applications, such as

simulations used in tactical planning tasks and exercises. The specific application that we plan to address is that of input of information for setting up simulations, though the generality of the representational issues will enable easy transfer to other tasks. The prototype will be demonstrated at the conclusion of Phase III.

## APPENDIX A
## CRITERIA FOR AVENUE OF APPROACH

### A. OBSERVATION AND FIRE

The highest point of a hill is not always the best place from which to observe; often the military crest is further down the hill due to the effect of vegetation and convex slope:

(1) Fields of fire are poor within wooded areas.

(2) Larger built-up areas provide poor observation and fields of fire.

(3) Smaller villages and farm buildings generally provide good observation and field of fire throughout the countryside.

### B. COVER AND CONCEALMENT

Concealment can be provided by underbrush, sand/snow drifts, tall grass, valleys, and cultivated vegetation.

Cover can be provided by rocks, ditches, qurries, caves, river banks, folds in ground, buildings, railroad embankments, sunken roads, backside of hill, and ravines.

Ridge provides little protection from enemy fire. Best axis is along ridge slopes that are below military crests rather than along valley floor.

Forest areas provide excellent cover from small-arms fire and concealment from air and ground observation (depending on tree type and season).

### C. OBSTACLES

Obstacles that are perpendicular to the avenue of approach provide an advantage for defense by slowing or canalizing the offense, but are a disadvantage for advancement.

Obstacles that are parallel to the avenue of approach provide an advantage to advancement and protecting flank of offense, but are a disadvantage to defense against attack.

(1) Natural Obstacles - variation in terrain such as rivers, swamps, marshes, forests, mountains, rock outcrops, soft soils, flooded ares, etc.

(2) Cultural Obstacles - built-up or urban areas, cities, buildings

(3) Reinforcing (Deliberate) Obstacles - added to terrain to strengthen existing obstacles or to create obstacles in open areas. Includes ditches, mines, craters, etc.

## D. ADEQUACY OF MANEUVER SPACE

Determines how restrictive the avenue is and where choke points are located.

Requirement for width of avenue of approach by unit:

| Unit | Width, km |
|---|---|
| Division | 6-10 |
| Brigade | 3-6 |
| Battalion | 1 |
| Company | 0.5 |

## E. EASE OF MOVEMENT

Concerned with the trafficabilty of avenue, relative length of candidate avenue compared to the avenues, directness of approach as means of gaining objective

Terrain can be classified in terms of trafficability as go, slow-go, or no-go:

| Speed, km/hr. | Manueverability |
|---|---|
| 32 - 40 | GO |
| 8 - 32 | SLOW GO |
| 0 - 8 | NO GO |
| Blocked | NO GO |
| Built-up | NO GO |

**1. Slope:**

(1) Tanks can handle slopes up to 45% and vehicles up to 1.5m

(2) Trucks can handles slopes up to 30% and verticles up to 1/3 of wheel height

**2. Soils:**

**(a) Gravel -**

1. Excellent for tracked vehicles when well-graded and compacted

2. May hamper wheeled vehicles

3. Independent of weather

(b) **Sand -**

1. Excellent trafficability when wet & compacted or mixed with clay

2. Obstacle to vehicles when very dry & loose; especially on slopes

(c) **Silt -**

1. Excellent trafficability, though dusty, when dry

2. Water causes softness & mud

(d) **Clay -**

1. Excellent trafficability when dry

2. Sticky & slippery with water obstacle when sloped

**3. Water bodies:**

Obstacles to cross-country movement except when sufficiently frozen Obstacle when:

(1) Crossings wider then 2.4m for medium tanks

(2) Fording 0.9-1.3m depths for tanks & 0.6-0.9m for trucks

(3) Verticle banks more than 1.3 for tanks, 0.3 for trucks

**4. Vegetation - Woods:**

(1) Provide good concealment and cover for elevated platforms with good fire

(2) Obstructs free passage & movement of armor & wheeled vehicles & slows dismounted troops

(3) Medium tanks push over single trees up to 20 cm in diameter and 2.5 ton trucks up to 5 cm

(4) Tanks & trucks can handle 4.5 - 6.5 m space between trees